

Remarks and Arguments

Claims 1-36 have been submitted for examination. Claims 1, 3, 11, 13, 21 and 23 have been amended. New claims 31-36 have been added.

Claims 1-30 have been rejected under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 6,263,379 (Atkinson) in view of U.S. Patent No. 6,460,058 (Koppolu.) The examiner comments that the Atkinson reference discloses the claimed subject matter except that Atkinson does not explicitly teach that the moniker object is substituted for the distributed object as claimed. However, the examiner asserts that the Koppolu reference discloses a stream object that substitutes the moniker object for the distributed object. The examiner concludes that it would have been obvious to combine the teachings of Atkinson and Koppolu because the combination provides seamless navigation between multiple document types as disclosed in Koppolu.

As discussed in the response to the previous office action, in the present invention, moniker objects assist in persisting and resurrecting distributed objects. In particular, during the streaming of the distributed object into local storage, the moniker object is automatically substituted for the actual object. Thus, the moniker object is stored in local storage in place of the real object. The real object is stored in a persistent repository in a server that can be remote to the local storage. The moniker object in the local storage can then be maintained by the inventive life cycle services system. This operation is described in detail in the present specification at page 22, line 3 to page 23, line 3 (in connection with figures 13 and 14.)

Similarly, when a distributed object is resurrected, the moniker object is instead streamed in from local storage. During this streaming, a reference to the real object is substituted for the moniker object in the memory. The real object is created in the remote server. This operation is described in detail in the present specification at page 23, lines 4-29 (in connection with figures 15 and 16.)

Applicant agrees with the examiner that the Atkinson reference does not teach such a substitution. However, the examiner contends that the Koppolu reference does teach such a substitution in the BindToObject and BindToStorage functions. The purpose of these latter functions is to allow a client to access diverse objects with the common interface of the moniker object. Thus, a client can use the moniker object to

manipulate other objects. It is clear from the Koppolu specification that the moniker object controls other objects, but it is not substituted for them. The real object is the object being stored and resurrected under control of the moniker object, the moniker object is not stored in the local storage in place of the real object. In column 15, lines 1-4, Koppolu describes the moniker object's BindToObject function that, as stated, instantiates a named object in the memory. However, this does not occur when the moniker object is retrieved from storage during the resurrection of the real object. This function allows an incoming data stream to instantiate an object directly in memory and is not involved with resurrection of an object from storage into memory.

Similarly, column 15, lines 4-8 of Koppolu describes the moniker object's BindToStorage function that, as stated, instantiates a named object into an OLE storage stream so that the named object is thereby created in the storage. This function allows an incoming data stream to be downloaded directly into storage as described in Koppolu.

The present claims have been amended to explicitly recite that the monitor object is substituted for the distributed object in the local storage during storage and resurrection operations. For example, claim 1, in lines 7-10, recites "a first stream object which automatically substitutes the moniker object for the distributed object during the streaming of the distributed object out from the memory to the local storage so that the moniker object is stored in the local storage in place of the distributed object." Neither of the cited references, nor their combination, teaches or suggests such a stream writer. The examiner points to Koppolu, column 14, line 64 to column 15, line 14 as disclosing a stream object that substitutes a moniker object for a distributed object when the distributed object is streamed from memory to local storage. However, in the section to which the examiner refers, Koppolu discloses three operations. First, in column 14, lines 55-63, Koppolu states that the moniker object contains stream functions that allow the moniker object to be streamed into and out of storage. Thus, the client can load and store the moniker object, but this process does not occur during a storage and resurrection of another object. The BindToObject and BindToStorage functions, discussed above allow the moniker object to download a data stream to either memory or to storage, but the moniker object is not substituted for another object in the

process as recited in amended claim 1. Consequently, amended claim 1 patentably distinguishes over the cited references.

Amended claims 11 and 21 contain parallel limitation to those noted above in amended claim 1 and patentably distinguish over the cited references in the same manner as amended claim 1.

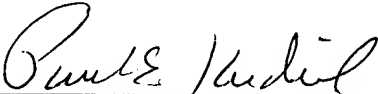
Claims 2-10 are dependent, either directly or indirectly, on amended claim 1 and incorporate the limitations thereof. Therefore, they also distinguish over the cited references in the same manner as claim 1. In addition, these claims also recite additional limitations not disclosed or suggested by Atkinson and Koppolu. For example, claim 2 recites that the moniker object is substituted for the distributed object when the distributed object is persisted. As stated above, the Koppolu moniker object persists the named object, not itself. Claim 3 has been amended to recite a second stream object which automatically substitutes a reference to the distributed object for the moniker object during the streaming of the moniker object in from the local storage to the memory so that a reference to the distributed object is created in memory in place of the moniker object. To the extent that Koppolu describes a substitution, this substitution does not occur when the moniker object is streamed into memory, but when the client calls the moniker object's BindToObject function. Further, in Koppolu, a reference to a real object is not created in place of the moniker object when the moniker object is retrieved from memory. Therefore, claim 2 and amended claim 3 patentably distinguish over the cited references.

Claims 12-20 are dependent, either directly or indirectly, on amended claim 11 and incorporate the limitations thereof. Therefore, they also distinguish over the cited references in the same manner as amended claim 11. In addition, these claims also recite limitations that parallel the limitations in claims 2-10 and thereby distinguish over the cited references in the same manner as claims 2-10.

Finally, claims 22-30 are dependent, either directly or indirectly, on amended claim 21 and incorporate the limitations thereof. Therefore, they also distinguish over the cited references in the same manner as amended claim 21. In addition, these claims also recite limitations that parallel the limitations in claims 2-10 and thereby distinguish over the cited references in the same manner as claims 2-10.

Applicant believes the claims are now in allowable condition. A notice of allowance for this application is solicited earnestly. If the examiner has any further questions regarding this amendment, he/she is invited to call applicant's attorney at the number listed below. The examiner is hereby authorized to charge any fees or credit any balances under 37 CFR §§1.17, and 1.16 to Deposit Account No. 09-0460.

Respectfully submitted



Date: 10/20/03

Paul E. Kudirka, Esq. Reg. No. 26,931
KUDIRKA & JOBSE, LLP
Customer Number 021127
Tel: (617) 367-4600 Fax: (617) 367-4656